Figure 1: A finite automaton $M_1$ (on left) and a finite automaton $M_2$ (on right).

1. Define:

$$A_{NFA} = \{ <B,w> \mid B \text{ is an NFA that accepts input string } w \}$$

$$EQ_{DFA} = \{ <B,C> \mid B, C \text{ are DFAs and } L(B) = L(C) \}.$$

Recall that DFA (respectively, NFA) stands for deterministic (respectively, nondeterministic) finite automaton. Answer the following questions (for $M_1$ and $M_2$ given in the above figure) and give reasons for your answers.

(a) Is $< M_1, babb > \in A_{NFA}$?

Yes. $M_1$ accepts babb.

(b) Is $< M_1, abba > \in A_{NFA}$?

No. $M_1$ does not accept abba.

(c) Is $< M_2, abb > \in A_{NFA}$?

Yes. $M_2$ accepts abb.

(d) Is $< M_2, aabb > \in A_{NFA}$?

Yes. $M_2$ accepts aabb.

(e) Is $< M_1, M_2 > \in EQ_{DFA}$?

No. $M_2$ is not a DFA.

(f) Is $< M_1, M_1 > \in EQ_{DFA}$?

Yes. $L(M_1) = L(M_1)$.

(g) Is $< M_2, M_2 > \in EQ_{DFA}$?

No. $M_2$ is not a DFA.
2. (a) Consider the language

\[ C = \{ < M, w > \mid M \text{ is a DFA, and some string of } L(M) \text{ contains string } w \text{ as a substring } \} \]

Is the language \( C \) decidable or undecidable? Prove your answer.

\[ \text{Decider for } C : \]

\[ P = " \text{On input } < M, w > \]

1. Construct a DFA \( A \) for \( L(M) \cap \Sigma^* w \Sigma^* \), where \( \Sigma \) is the alphabet of \( M \). This is possible because regular languages are closed under intersection.
2. Decide whether \( L(A) = \emptyset \) (DFA emptiness is decidable)
   - If yes, reject. If no accept"

(b) Consider the language

\[ D = \{ < M > \mid M \text{ is a Turing machine and there exists a DFA } A \text{ such that } L(M) = L(A) \} \]

Is the language \( D \) decidable or undecidable? Prove your answer.

\[ D \text{ is undecidable. Two justifications (only one needed) } \]

i) \( D \) is a non-trivial semantic property. Undecidability follows from Rice's theorem.

ii) \( D \) encodes the problem of deciding whether the language recognized by a TM is regular. This is shown in the text to be undecidable (in Theorem 5.3)
3. Give an implementation-level description of a deterministic one tape Turing machine that decides the following language $A$ over the alphabet $\Sigma = \{c, d\}$. The number of occurrences of symbol $c$ (respectively, $d$) in a string $w$ is denoted $|w|_c$ (respectively, $|w|_d$).

$$A = \{ w \in \Sigma^* \mid |w|_c \leq |w|_d \leq 2 \cdot |w|_c \}.$$  

That is, in strings of $A$ the number of occurrences of $d$ is at least the number of occurrences of $c$ and at most 2 times the number of occurrences of $c$.

$P =$ "On input $w \in \Sigma^*$:

1. Repeat the following as long as both $c$'s and $d$'s remain:
   - scan the tape and mark one $c$ and one $d$.
2. If after 1. there remain $c$'s on the tape reject.
   Else continue from 3. (unmarked)
3. Unmark all $c$'s and $d$'s.
4. Repeat the following as long as both unmarked $c$'s and unmarked $d$'s remain:
   - scan the tape and mark two $d$'s and one $c$.
5. If after the last scan unmarked $d$'s remain, reject.
   Else accept /* this includes the case where the last scan could mark only one $d$ */"
4. Let

$$\text{TWO}_\text{TM} = \{ < M > \mid M \text{ is a deterministic Turing machine and}$$

$$L(M) \text{ consists of exactly two strings } \}.$$ 

Without using Rice's theorem show that $\text{TWO}_\text{TM}$ is undecidable.

We reduce $A_{\text{TM}}$ to $\text{TWO}_\text{TM}$. Suppose $\text{TM } P$ decides $\text{TWO}_\text{TM}$. Construct decide $Q$ for $A_{\text{TM}}$:

$$Q = \text{"On input } < M, w > \text{"}$$

1. Construct decide $\text{TM } N$:

$$N = \text{"On input } x \text{"}$$

   1. If $x = \alpha$ or $x = \alpha \alpha$ ($\alpha \in \Sigma$), accept.

   2. Simulate $M$ on $w$. If $M$ accepts, accept.
      If $M$ rejects, reject.

2. Run $P$ on $< N >$. If $P$ accepts, reject.
   If $P$ rejects, accept.

Explanation why this works:

If $< M, w > \notin A_{\text{TM}}$, then $L(N) = \{ \alpha, \alpha \alpha \}$ and $< N > \notin \text{TWO}_\text{TM}$.
If $< M, w > \in A_{\text{TM}}$, then $L(N) = \Sigma^*$ and $< N > \notin \text{TWO}_\text{TM}$.
(a) (7 marks) Let \( T = \{ (i, k, m) \mid i, k, m \in \mathbb{N} \} \).
Show that the set \( T \) is countable.

All elements of \( T \) can be listed as follows:

1. List \((1,1,1)\)
2. List triples \((i,k,m)\) where \(i+k+m = 4\)
3. List triples \((i,k,m)\) where \(i+k+m = 5\)

Stage \( x \): List triples \((i,k,m)\) where \(i+k+m=x+2\)

All elements of \( T \) appear on above list.

(b) (3 marks) Are the following sets countable? For each case circle the correct answer
- no explanation needed. If you circle both YES and NO, it is considered a wrong answer.

- The set \( \{0,1\}^* \).  YES  NO

- The set of all subsets of \( \{0,1\}^* \).  YES  NO

- The set of all finite subsets of \( \{0,1\}^* \).  YES  NO